Amendments to Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims

1. (Currently Amended) A system that converts words of pipelined data from a first data width to a smaller second data width over a series of cycles, comprising:

a pipeline control that generates a word output from the pipeline control that is at least twice as wide as the ~~smallest~~ first data width entering the pipeline control, and shifts the pipelined data within the generated word such that the pipelined data word is, depending on the cycle, either prefixed by zeros, suffixed by zeros, or both, and that outputs a residual portion that is at least as wide as the ~~smaller~~ word output having a second data width and a current portion that is at least as wide as the ~~smaller~~ second data width;

a delay register that receives at least a portion of the residual portion of the data from the pipeline control and delays the received residual portion one cycle from the current portion of the data from the pipeline control;

a ~~combinor~~ combiner circuit that compares the delayed residual portion of data from the delay register with the current portion of data from the pipeline control, and outputs data having a width equal to the ~~smaller~~ second data width; and

output conduits that, for every complete series, transport;

the current portion of the data during the initial cycle;

the ~~combinor~~ combiner circuit output during the non-initial cycles; and

the residual portion of the data during the final cycle.

2. (Original) The system of claim 1, wherein:

the first data width is 66 bits;

the second data width is 64 bits; and

the series repeats itself every 32 cycles.

3. (Original) The system of claim 1, wherein the system is used in an interface circuit.

4. (Currently Amended) The system of claim 1, wherein:

the cycle is identified by a cycle value; and

the pipeline control includes:

a plurality of shift control circuits for ~~a~~ shifting the pipelined data, each shift control circuit shifting the data a different amount and being controlled by the cycle value.

5. (Currently Amended) The system of claim 1, wherein:

the cycle is identified by a cycle value; and

the pipeline control includes:

a plurality of shift control circuits for ~~a~~ shifting the pipelined data, each shift control circuit shifting the data a different amount and being controlled by the cycle value~~, the amount being shifted being an exponential function~~.

6. (Currently Amended) The system of claim 1, further comprising a ~~multiplexor~~ multiplexer that, in the initial cycle of every series, disregards the result from the ~~combiner~~ combiner circuit and directs the current portion of the data to the output conduit.

7. (Currently Amended) The system of claim 1, further comprising a zero-loading circuit that, in the initial cycle of every series, replaces the inputs to the ~~combiner~~ combiner circuit with the current portion of data from the pipeline control and zeros such that the output will equal the current portion of the data~~, and can be directed to the output conduit~~.

8. (Currently Amended) A system that converts pipelined data from a first data width to a smaller second data width over a series of cycles, comprising:

a pipeline control that, depending on the cycle, outputs a residual portion and a current portion;

a delay register that receives the residual portion of the data from the pipeline control and delays the received residual portion one cycle from the current portion of the data from the pipeline control;

a ~~combiner~~ combiner circuit that compares the delayed residual portion of data from the delay register with the current portion of data from the pipeline control, and outputs data; and

a flow-through logic circuit that outputs data having a width ~~that is at least~~ as wide as the ~~smaller~~ second data width from the pipelined data in the first cycle in which data is outputted ~~of~~ every series of cycles.

9. (Currently Amended) A transmitting interface comprising:

a system that encapsulates a data stream and prepares the encapsulated data stream for transmission over an optical network, including:

a system that encapsulates the data stream;

a system that ~~system that~~ converts the encapsulated data stream from a first data width to a smaller second data width over a series of cycles, including:

a pipeline control that, depending on the cycle, shifts encapsulated data and outputs a residual portion that is at least as wide as the ~~smaller~~ second

data width and a current portion that is at least as wide as the ~~smaller~~ second data width;

a delay register that receives at least a portion of the residual portion of the data from the pipeline control and delays the received residual portion one cycle from the current portion of the data from the pipeline control;

a ~~combiner~~ combiner circuit that compares the delayed residual portion of data from the delay register with the current portion of data from the pipeline control, and outputs data having a width equal to the ~~smaller~~ second data width; and

a flow-through logic circuit that outputs data having a width that is at least as wide as the ~~smaller~~ second data width from the pipelined data in the first cycle in which data is outputted ~~of~~ every series of cycles; and

a system that prepares outputted data for conversion to optical signals;

a system that serializes data streams that are encapsulated and prepared for conversion to optical signals; and

a system that transmits optical signals.

10. (Currently Amended) A method of converting a series of input data words of a first data width to a series of output data words of a smaller second data width, concatenation of the series of input data being equivalent to concatenation of the series of output data, the series having a sequence including an initial sequence, a non-initial sequence, a final sequence and a non-final sequence, comprising:

identifying a first portion and a second portion of each input data word, the size of each portion depending on the progression of the input series, the size of the first portion initially having a size equal to the ~~smaller~~ second data width, gradually decreasing as the input series progresses, and the size of the second portion gradually increasing until the size is equal to the ~~smaller~~ second data width when the input series is completed;

delaying the second portion of each input data word for at least the non-final sequences of each input series;

combining the delayed second portion with the first portion for at least the non-initial sequences of each input series;

outputting the first portion for the initial sequence of each output series;

outputting the combination ~~for at least the non-initial and non-final sequences of each output series~~ during the interim between the initial sequence and the final sequence; and

outputting the second portion for the final sequence of each completed output series.

11. (Original) The method of claim 10, wherein:

the first data width is 66 bits;

the second data width is 64 bits; and

each series has a maximum of 32 increments of ordered data words.

12. (Original) The method of claim 11, wherein the size of the first portion decreases by 2 bits and the size of the second portion increases by 2 bits for every increment.

13. (Original) The method of claim 12, wherein the system is used in an interface circuit.